PR4

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/038,203	10/19/2001	Krishneadu Chakraborty	0007056-0226/P6339	8440

7590 07/17/2003

SONNENSCHEIN NATH & ROSENTHAL
P.O. BOX 061080
WACKER DRIVE STATION, SEARS TOWER
CHICAGO, IL 60606-1080

EXAMINER	
MAHMOUDI, HASSAN	
ART UNIT	PAPER NUMBER
2175	3

DATE MAILED: 07/17/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PR4

Office Action Summary	**Application No.** 10/038,203	**Applicant(s)** CHAKRABORTY ET AL.
	Examiner Tony Mahmoudi	**Art Unit** 2175

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____ .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-36 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-36 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

DOV POPOVICI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .

4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: .

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-2, 6-11, 15-20, 24-29, and 33-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pearson (U.S. Pub. No. 2003/0061191) in view of Larsen (U.S. Pub. No. 2003/0055850.)

As to claim 1, Pearson teaches a method (see page 1, paragraph 15) for merging (see page 1, paragraph 2) one or more hierarchical trees (see Abstract), comprising:

examining one or more nodes in each of the hierarchical trees (see page 2, paragraph 24);

determining if there are one or more sets of equivalent nodes in the hierarchical trees (see page 2, paragraph 25, where "equivalent nodes" is read on "duplicate");

picking one or more winning nodes from each of the sets of equivalent nodes (see page 2, paragraph 26); and

storing one or more reference nodes (see page 1, paragraph 16, and see page 4, paragraph 46) to the winning nodes (see page 4, paragraph 47.)

Pearson does not teach merging at run time.

Larsen teaches a method for dynamically generating a list of items (see Abstract), in which he teaches merging at run time (see page 1, paragraphs 5 and 6.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson to include merging at run time.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson by the teaching of Larsen, because merging at run time, would enable the system to merge data nodes of trees only when the desired function is being executed (e.g. in an on-line HELP environment, the related data nodes are merged together only at the time HELP function is executed.), as taught by Larsen (see page 1, paragraph 5.)

As to claims 2, 11, 20, and 29, Pearson as modified teaches wherein the hierarchical trees are document object model (DOM) trees (see Larsen, page 2, paragraph 22.)

As to claims 6, 15, 24, and 33, Pearson as modified teaches wherein the picking further comprises:

examining one or more priorities associated with one or more members in each set of the equivalent nodes (see Pearson, page 2, paragraphs 24 and 25, where "priority" is read on "key value"); and

selecting the winning node as the member with a highest of the priorities (see Pearson, page 2, paragraph 26.)

As to claims 7, 16, 25, and 34, Pearson as modified teaches further comprising generating one or more shallow clones for the winning nodes; and adding the shallow clones to a merged tree (see Pearson, page 3, paragraphs 38-40, and see pages 4-5, paragraph 50.)

As to claims 8, 17, 26, and 35, Pearson as modified teaches wherein the hierarchical trees include a group tree, a user tree, and an admin tree (see Pearson, figures 5A through 5L. where multi-level hierarchical trees are shown. It is inherent that in a hierarchical tree structure, the tree consists of multiple levels, i.e. "group level", "user level", and "admin level", which in the referenced figures, can be depicted in any of the hierarchical levels 0-5.)

As to claims 9, 18, 27, and 36, Pearson as modified teaches wherein the DOM trees (see Larsen, page 2, paragraph 22) are eXtensible Markup Language (XML) DOM trees (see Larsen, page 2, paragraph 21.)

As to claim 10, Pearson teaches a system (see Abstract) for merging (see page 1, paragraph 2) one or more hierarchical trees comprising:

one or more nodes in each of the hierarchical trees configured to be examined (see page 2, paragraph 24);

one or more sets of equivalent nodes in the hierarchical trees configured to be located if the sets of equivalent nodes exist (see page 2, paragraph 25, where "equivalent nodes" is read on "duplicate");

one or more winning nodes configured to be picked from each set of the equivalent nodes (see page 2, paragraph 26); and

one or more reference nodes to the winning nodes configured to be stored (see page 1, paragraph 16, and see page 4, paragraph 46) to the winning nodes (see page 4, paragraph 47.)

Pearson et al does not teach merging at run time.

Larsen teaches a method for dynamically generating a list of items (see Abstract), in which he teaches merging at run time (see page 1, paragraphs 5 and 6.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson to include merging at run time.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson by the teaching of Larsen, because merging at run time, would enable the system to merge data nodes of trees only when the desired function is being executed (e.g. in an on-line HELP environment, the related data nodes are merged together only at the time HELP function is executed.), as taught by Larsen (see page 1, paragraph 5.)

As to claim 19, Pearson teaches a computer program product (see page 1, paragraph 2) comprising:

a computer usable medium (see page 1, paragraph 16) having computer readable program code embodied therein (it is inherent that computers are programmed with "computer readable program code") configured to merge one or more hierarchical trees (see page 1, paragraph 2) comprising:

computer readable code configured to cause a computer to examine one or more nodes in each of the hierarchical trees (see page 2, paragraph 24);

computer readable code configured to cause a computer to determine if there are one or more sets of equivalent nodes in the hierarchical trees (see page 2, paragraph 25, where "equivalent nodes" is read on "duplicate");

computer readable code configured to cause a computer to pick one or more winning nodes from each set of the equivalent nodes (see page 2, paragraph 26); and

computer readable code configured to cause a computer to store one or more reference nodes (see page 1, paragraph 16, and see page 4, paragraph 46) to the winning nodes (see page 4, paragraph 47.)

Pearson et al does not teach merging at run time.

Larsen teaches a method for dynamically generating a list of items (see Abstract), in which he teaches merging at run time (see page 1, paragraphs 5 and 6.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson to include merging at run time.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson by the teaching of Larsen, because merging at run time, would enable the system to merge data nodes of trees only when the desired function is being executed (e.g. in an on-line HELP environment, the related data nodes are merged together only at the time HELP function is executed.), as taught by Larsen (see page 1, paragraph 5.)

As to claim 28, Pearson teaches an apparatus (see pages 1-2, paragraph 17) for merging (see page 1, paragraph 2) one or more hierarchical trees comprising:

means (see pages 1-2, paragraph 17) for examining one or more nodes in each of the hierarchical trees (see page 2, paragraph 24);

means (see pages 1-2, paragraph 17) for locating one or more sets of equivalent nodes in the hierarchical trees, if the sets of equivalent nodes exist (see page 2, paragraph 25, where "equivalent nodes" is read on "duplicate");

means (see pages 1-2, paragraph 17) for picking one or more winning nodes from each set of the equivalent nodes (see page 2, paragraph 26); and

means (see pages 1-2, paragraph 17) for storing one or more reference nodes (see page 1, paragraph 16, and see page 4, paragraph 46) to the winning nodes (see page 4, paragraph 47.)

Pearson et al does not teach merging at run time.

Larsen teaches a method for dynamically generating a list of items (see Abstract), in which he teaches merging at run time (see page 1, paragraphs 5 and 6.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson to include merging at run time.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson by the teaching of Larsen, because merging at run time, would enable the system to merge data nodes of trees only when the desired function is being executed (e.g. in an on-line HELP environment, the related data nodes are merged together only at the time HELP function is executed.), as taught by Larsen (see page 1, paragraph 5.)

3. Claims 3, 12, 21, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pearson (U.S. Pub. No. 2003/0061191) in view of Larsen (U.S. Pub. No. 2003/0055850), as applied to claims 1-2, 6-11, 15-20, 24-29, and 33-36 above, and further in view of Geil (U.S. Patent No. 3,662,400.)

As to claims 3, 12, 21, and 30, Pearson as modified does not teach:

printing a merged tree using the reference nodes.

Geil teaches a subsidiary document identification system (see Abstract), in which he teaches printing a merged tree using the reference nodes (see column 10, lines 45-60.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson as modified to include printing a merged tree using the reference nodes.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson as modified, by the teaching of Geil, because printing a merged tree using the reference nodes, would enable the user to obtain a copy of the properly arranged, tier-oriented detail merged tree, as taught by Geil (see column 10, lines 65-67.)

4. Claims 4-5, 13-14, 22-23, and 31-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pearson (U.S. Pub. No. 2003/0061191) in view of Larsen (U.S. Pub. No. 2003/0055850), as applied to claims 1-2, 6-11, 15-20, 24-29, and 33-36 above, and further in view of Blais et al (U.S. Pub. No. 2002/0178437.)

As to claims 4, 13, 22, and 31, Pearson as modified does not teach wherein the reference nodes are one or more pointers.

Blais et al teaches an object-oriented allocation method and apparatus (see Abstract), in which he teaches wherein the reference nodes are one or more pointers (see pages 9-10, paragraphs 107 through 109.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson as modified to include wherein the reference nodes are one or more pointers.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson as modified, by the teaching of Blais et al, because the reference nodes being one or more pointers, would enable the system to identify the selected nodes in order to merge the nodes together, to form a newly merged tree.

As to claims 5, 14, 23, and 32, Pearson as modified does not teach wherein the reference nodes are one or more Java references.

Blais et al teaches an object-oriented allocation method and apparatus (see Abstract), in which he teaches wherein the reference nodes are one or more Java references (see page 4, paragraph 66.)

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson as modified to include wherein the reference nodes are one or more Java references.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Pearson as modified, by the teaching of Blais et al, because the reference nodes being one or more Java references, would provide an object-oriented environment (Java environment) for referencing memory and accessing objects without explicitly checking the object.

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The following patents are cited to further show the state of art with respect to methods and systems of merging hierarchical trees within databases in general:

Patent No.	**Issued to**	**Cited for teaching**
US 5,970,490	Morgenstern	Data processing and merging trees.
US 6,094,664	Ungar	Object-oriented programs and Java references
US PUB. 2002/0083073	Vaidya et al.	Managing layered hierarchical data sets in tree structures.

6. Any inquiries concerning this communication or earlier communications from the examiner should be directed to Tony Mahmoudi whose telephone number is (703) 305-4887. The examiner can normally be reached on Mondays-Fridays from 08:00 am to 04:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Dov Popovici, can be reached at (703) 305-3830.

tm

July 11, 2003

DOV POPOVICI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100